



Classification Analysis with Python Theory

This documents briefly covers the basics concepts of classification analysis

- The target/dependent variable is a discrete variable whereas the input variables have any measurement level.
- Questions that can be answered through classification analysis:
 - Will it be sunny or not?
 - Will the customer default or not?
 - Should we invest or not?
- **Types of Classification**
 1. **Binary Classification**
 - We group an outcome into two groups i.e. Male - Female, Yes - No, Pass - Fail etc.
 - The data is usually represented as 0's and 1's in our data.
 2. **Multi-class Classification**
 - We group an outcome into more than two groups i.e. Yes - No - Maybe, Pass - Fail - Unclassified, etc.
 3. **Ordinal Logistic Regression**
 - We group an outcome into more than two groups with ordering i.e. low, medium, high.
- Steps for building a classifier:
 1. Business Understanding
 2. Data Understanding
 3. Data Preparation
 4. Data Modeling
 - Splitting the dataset
 - Building a classifier
 - Model evaluation
 - Model optimisation
 5. Model Deployment / Presentation
- Classification algorithms include:
 1. **Logistic Regression**
 2. **Naive Bayes**
 3. **K-Nearest Neighbors**
 4. **Support Vector Machine**
 5. **Decision Tree**
 6. **XGBoost**

7. MLP (Multi-layer Perceptron) Neural network

Logistic Regression Classification

- Predicted values are the probability of a particular level of the target variable at the given values of the input variables.
- We use the logistic function/ sigmoid function output a value between 0 and 1. This would be the predicting values to probabilities.
 - $S(z) = 1/(1+e^{-z})$
- We then select a line (decision boundary) that depends on the use case. Any data point with a probability value above the line is classified into the class represented by 1. The data point below the line is classified into the class represented by 0.
- Assumptions:
 - Binary output: Ensure output variable values are classified into 0 or 1.
 - Noise Removal: Remove any outliers or misclassified values in the dataset.
 - Normal Distribution: Ensure normal /gaussian distribution is present across individual inputs.
 - Multicollinearity: Dealing with highly correlated inputs.

Naive Bayes Classification

- Applies the Bayes' theorem to calculate the probability of a data point belonging to a particular class.
- Given the probability of certain related values, the formula to calculate the probability of an event B, given event A to occur is calculated as follows.
 - $P(B|A) = P(A|B) * P(B) / P(A)$
- Types of Naive Bayes:
 - Gaussian Naive Bayes
 - Multinomial Naive Bayes
 - Bernoulli Naive Bayes
- Naive Bayes algorithms are fast, highly scalable and can easily train smaller dataset. They do well for multi - class prediction problems.
- Assumption
 - Multicollinearity: It assumes that there is no dependency between any of the input features.

K-Nearest Neighbors (KNN) Classification

- K-nearest neighbors (KNN) takes the approach that data points are considered to belong to the class with which it shares the most number of common points in terms of its distance.
- Steps of KNN:
 - Calculate the distance between any two points.
 - The most popular formula to calculate this is the Euclidean distance.

- Find the nearest neighbours based on these pairwise distances.
 - Majority vote on a class labels based on the nearest neighbour list.
- It is a lazy learning algorithm because it does not have a specialised training phase and uses all the data for training while performing classification.
- KNN is very simple to implement but computationally expensive as it high memory is required during training.

Support Vector Machine (SVM) Classification

- Support Vector Machine algorithms take the approach where there is an output of an optimal line (also known as a hyperplane) of separation between the classes, based on the training data entered as input.
- This approach takes into consideration outliers that lie close to another class to derive this separating hyperplane. After the model is constructed with this hyperplane, any new point to be predicted checks to see which side of the hyperplane this value lies in.
- In addition, while performing classification with SVM, data is normally mapped to a higher dimensional space to create the above-mentioned separation. The mapping is done through the use of kernel functions.
- The commonly used kernel functions include:
 - The gaussian radial basis function (RBF)
 - The polynomial function.
- SVM is effective with high dimensional data i.e. high features: observation ratio has less impact with data with outliers and suited for an extreme case of binary classification. However, data needs to be normalised first. During normalisation, data is scaled to eliminate redundancy.

Decision Trees Classification

- This type of classification takes a dataset, breaks it into smaller subsets through the use of splitting rules to predict an output. These splitting rules are normally dependent on the patterns found in the dataset.
- The use of decision tree algorithms such as ID3, C4.5 and CART develops reasonably accurate decision trees, in a reasonable amount of time, employing a greedy strategy that grows a decision tree by making a series of locally optimum decisions.
- Splitting measures such as Gini Index, Information Gain etc. are used to perform the best split at every node of the decision tree.
- Decision trees classification is fast and requires less effort training thus computationally less expensive.

Classification Evaluation

- There are many ways of evaluating the performance of a classifier i.e. classification accuracy, confusion matrix, logarithmic loss, area under roc curve, etc.
- Classification Report
 - The precision will be "how many are correctly classified among that class".
 - The recall means "how many of this class you find over the whole number of element of this class".
 - The f1-score is the harmonic mean between precision & recall.
 - The support is the number of occurrence of the given class in your dataset.
- Classification Accuracy
 - This is the number of correct predictions divided by all predictions or ratio of correct predictions to total predictions.
 - Often used to used when the no. of observations in each class is roughly equivalent.
- Confusion Matrix
 - This is a table or chart representing the accuracy of a model with regards to two or more classes.
 - The predictions of the model will be on the x-axis while the accuracy is located on the y-axis.
 - Correct predictions can be found on a diagonal line moving from the top left to the bottom right.
- Logarithmic Loss
 - Evaluates how confident a classifier is about its predictions.
 - The value for predictions run from 1 to 0 with 1 being completely confident and 0 being no confidence.
 - The overall lack of confidence/loss is returned as a negative number with 0 representing a perfect classifier.

References

- Overview of Classification methods in Python with Scikit - Learn.
 - <https://stackabuse.com/overview-of-classification-methods-in-python-with-scikit-learn/>
- Learning Classification Algorithms
 - <https://developer.ibm.com/technologies/data-science/tutorials/learn-classification-algorithms-using-python-and-scikit-learn/>